AMENDMENTS TO THE CLAIMS

1.   (Currently amended) In a computer system, a method for alerting one or more computer software application threads waiting to retrieve events from an event port, the method comprising:

receiving, at the event port, an alert event generated by a computer software application;

changing a state of the event port to an alert state, if the event port is not already in an alert state, in response to receiving the alert event; [[and]]

notifying one or more of the computer software application threads about the alert state of the event port; and

returning at least one computer software application thread to its respective computer software application independent of each event that the at least one computer software application thread is waiting to retrieve.

2.   (Original) The method of claim 1, further comprising:

retrieving the alert event from the event port with the notified one or more computer software application threads; and

returning each computer software application thread to its respective computer software application with the retrieved alert event.

3.   (Original) The method of claim 1, wherein:

the alert event includes one or more flags; and

changing includes determining whether the event port is in an alert state by checking one or more of the flags.

4.   (Original) The method of claim 1, further comprising:

changing the event port from the alert state to a normal state

5. (Original) The method of claim 4, further comprising:

generating an error message if the event port cannot be changed to the normal state.

6. (Original) The method of claim 1, further comprising:

generating an error message if the event port cannot be changed to the alert state.

7. (Original) The method of claim 6, wherein the error message is generated in response to detecting one or more of:

an invalid port identifier, an event port argument not being an event port file descriptor, an event port already being in alert mode, and mutually exclusive flags being set.

8. (Original) The method of claim 1, further comprising:

including in the alert event data about the cause of the alert.

9. (Original) The method of claim 1, further comprising:

including in the alert event a reference to data about the cause of the alert.

10. (Original) The method of claim 2, wherein returning comprises:

returning each computer software application thread to its respective computer software application with the retrieved alert event and information about the cause of the alert event.

11. (Original) The method of claim 1, wherein the alert event is generated in response to one or more of the following actions:

a signal occurring, a synchronization request being issued, a task waiting to be performed, and a command being issued for terminating all ongoing processes.

12. (Original) The method of claim 1, wherein the event port has an associated event queue, further comprising:

placing the alert event in the event queue; and

keeping the alert event in the event queue until a request to remove the alert event is received.

13.   (Currently amended) A computer program product, stored on a machine-readable

medium, comprising instructions operable to cause a computer to:

receive, at the event port, an alert event generated by a computer software

application;

change a state of the event port to an alert state, if event port is not already in an alert

state, in response to receiving the alert event; [[and]]

notify one or more ~~of the~~ computer software application threads <u>waiting to retrieve events</u>

<u>from an event port</u> about the alert state of the event port<u>; and</u>

<u>return at least one computer software application thread to its respective computer software</u>

<u>application independent of each event that the at least one computer software application thread is</u>

<u>waiting to retrieve.</u>

14.   (Original) The computer program product of claim 13, further comprising

instructions to:

retrieve the alert event from the event port with the notified one or more computer software

application threads; and

return each computer software application thread to its respective computer software

application with the retrieved alert event.

15.   (Original) The computer program product of claim 13, wherein:

the alert event includes one or more flags; and

the instructions to change include instructions to determine whether the event port is in an

alert state by checking one or more of the flags.

16.   (Original) The computer program product of claim 13, further comprising

instructions to:

change the event port from the alert state to a normal state

17.   (Original) The computer program product of claim 16, further comprising
instructions to:

    generate an error message if the event port cannot be changed to the normal state.

18.   (Original) The computer program product of claim 13, further comprising
instructions to:

    generate an error message if the event port cannot be changed to the alert
state.

19.   (Original) The computer program product of claim 18, wherein the error message is
generated in response to detecting one or more of:

    an invalid port identifier, an event port argument not being an event port
file descriptor, an event port already being in alert mode, and mutually exclusive flags being set.

20.   (Original) The computer program product of claim 13, further comprising
instructions to:

    include in the alert event data about the cause of the alert.

21.   (Original) The computer program product of claim 13, further comprising
instructions to:

    include in the alert event a reference to data about the cause of the alert.

22.   (Original) The computer program product of claim 14, wherein the instructions to return
comprises instructions to:

    return each computer software application thread to its respective computer software
application with the retrieved alert event and information about the cause of the alert event.

23.   (Original) The computer program product of claim 13, wherein the alert event is
generated in response to one or more of the following actions:

a signal occurring, a synchronization request being issued, a task waiting to be performed, and a command being issued for terminating all ongoing processes.

24.   (Original) The computer program product of claim 13, wherein the event port has an associated event queue, further comprising instructions to:

place the alert event in the event queue; and

keep the alert event in the event queue until a request to remove the alert event is received.

25.   (Currently amended) An alert management apparatus for alerting one or more computer software application threads waiting to retrieve events from an event port in a computer system, comprising:

an event queue for receiving transaction events generated by one or more event sources, the event queue being accessible through an event port;

a request queue for holding requests to retrieve transaction events from the event queue, each request having an associated priority determining a place of the request in the request queue; and

a queue manager operable to:

receive, at the event port, an alert event generated by a computer software application;

change a state of the event port to an alert state, if the event port is not already in an alert state, in response to receiving the alert event; [[and]]

notify one or more of the computer software application threads about the alert state of the event port; and

return at least one computer software application thread to its respective computer software application independent of each event that the at least one computer software application thread is waiting to retrieve.